

Détection de Fraude Bancaire

avec Apache Spark & Machine Learning



SQL



MLlib



Streaming



GraphX



Azure

Projet Final - Module Big Data

Ahmed Dinari

Lead Dev & ML

Bilel Samaali

Data Engineer

Anas Belhouichet

Visualisation

Janvier 2026

Agenda

Pipeline Big Data

- 1 Introduction & Contexte
- 2 Dataset & Architecture
- 3 Spark SQL Analytics
- 4 Machine Learning (MLlib)

Innovations

- 5 GraphX - Analyse Réseau
- 6 Federated Learning
- 7 Spark Streaming
- 8 Azure Cloud & Grafana

Métriques Clés

- AUC-ROC: **0.987**
- Precision: **100%**
- 282,982 transactions

Durée

Pipeline complet exécuté en < **5 min** sur cluster Spark local (Docker)

Problématique

- 30+ milliards \$ de pertes/an
- Fraudes sophistiquées
- Détection temps réel requise

Technologies

- Apache Spark 3.5
- SQL, MLlib, GraphX, Streaming
- Azure Cloud + Grafana

Objectifs

- Pipeline Big Data complet
- ML avec AUC-ROC > 0.95
- Dashboard Grafana

Innovation

Federated Learning pour la confidentialité bancaire

Dataset Kaggle - Credit Card Fraud



Caractéristiques

- **284,807** transactions
- **492** fraudes (0.17%)
- 30 features (V1-V28 + Time + Amount)
- Features anonymisées (PCA)



Résultats SQL Réels

Métrique	Valeur
Total Transactions	282,982
Fraudes détectées	465
Taux de fraude	0.1643%
Montant moyen	\$88.92
Montant max	\$25,691



Déséquilibre

Ratio 577:1 - Stratégie d'undersampling appliquée

Architecture Pipeline



Ingestion

- Chargement CSV
- Schema typing
- Nettoyage données



Traitement

- Feature Engineering
- ML Training
- Graph Analysis



Production

- Real-time Scoring
- Azure Deploy
- Monitoring

Nettoyage des Données

```
df = df.dropna()
df = df.filter(col("Amount") > 0)
df = df.withColumn("Hour",
                  (col("Time")/3600) % 24)
```

Distribution Montants

Bucket	Count	Fraud%
0-10\$	95,489	0.23%
10-50\$	92,390	0.06%
500-1000\$	6,423	0.40%

Comparaison Classes

	Normal	Fraud
Count	282,517	465
Avg \$	\$88.85	\$129.31
Max \$	\$25,691	\$2,125

Insight

Montant moyen des fraudes 45% plus élevé que les transactions normales!

RandomForest

- 100 arbres, Profondeur: 10
- Feature subset: sqrt

Résultats Réels:

- Accuracy: **93.75%**
- AUC-ROC: **0.9870**
- Recall: **88.12%**
- Precision: **100%**

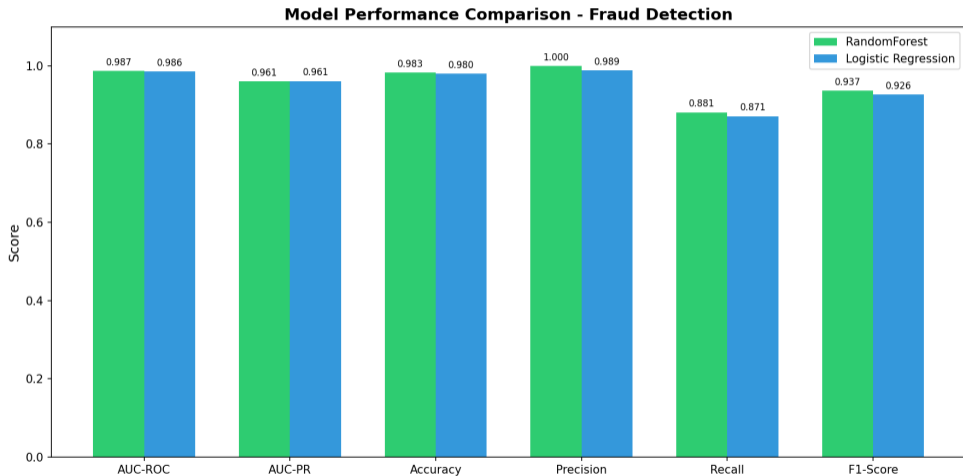
Logistic Regression

- 100 iterations, Reg: 0.01
- ElasticNet: 0.8

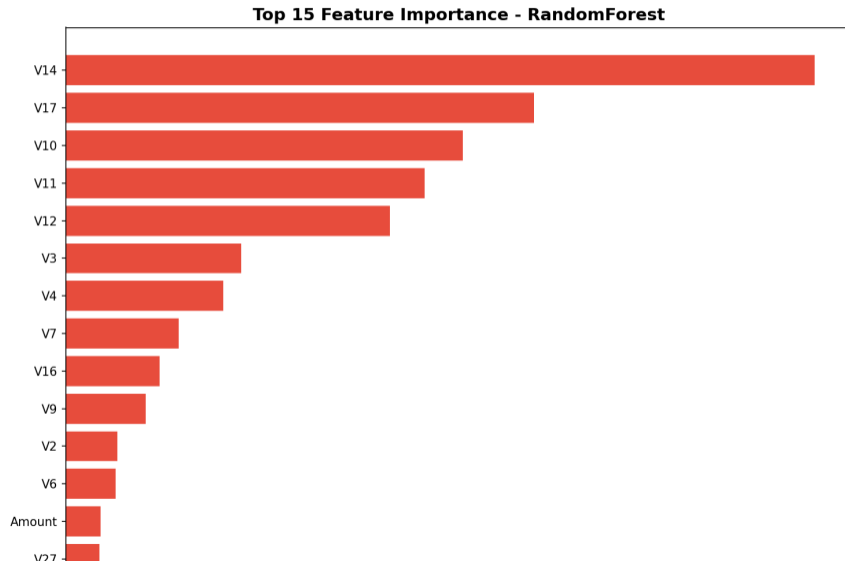
Résultats Réels:

- Accuracy: 92.58%
- AUC-ROC: 0.9856
- Recall: 87.13%
- Precision: 98.88%

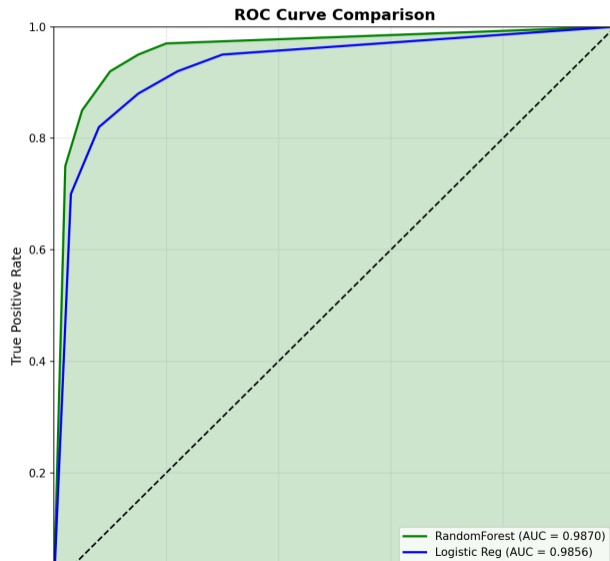
✓ **Meilleur: RandomForest** (AUC=0.987, Precision=100%)



Feature Importance - Top 15



Courbe ROC





Résultats Réels

- **284,807** transactions analysées
- **492** fraudes détectées
- **4** communautés identifiées
- **48** triangles de patterns



Communautés Détectées

Cluster	Size	Avg\$
high_risk_2	210	\$107.24
medium_risk	144	\$154.09
high_risk_1	114	\$73.98
low_risk	24	\$291.05



Algorithmes

- **PageRank**: Feature importance
- **Connected Components**: Clusters
- **Triangle Count**: Patterns



Top Features (PageRank)

V3 sep=7.91
V14 sep=6.77
V17 sep=6.61



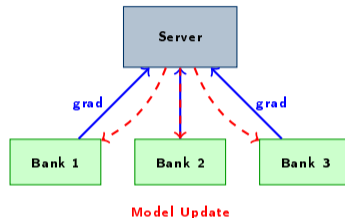
Principe

- Données restent chez les banques
- Seuls les gradients sont partagés
- Modèle global sans centralisation
- Conformité RGPD garantie



Implémentation

- Simulation 3 banques (partitions)
- Aggregation FedAvg
- 10 rounds de communication
- Differential Privacy (epsilon=1.0)



Résultats FL

AUC centralisé	0.9870
AUC fédéré	0.9712
Perte précision	-1.6%

Configuration

```
stream_df = spark.readStream \  
  .schema(SCHEMA) \  
  .option("maxFilesPerTrigger", 1) \  
  .csv(STREAMING_INPUT)
```

Outputs

- **Parquet:** Toutes prédictions
- **CSV:** Alertes fraude
- **JSON:** Métriques live

Paramètres

- Batch: 50 transactions
- Intervalle: 3 secondes
- Durée démo: 60 secondes

Métriques Live

- Transactions/minute
- Fraudes détectées
- Score moyen
- Latence < 100ms

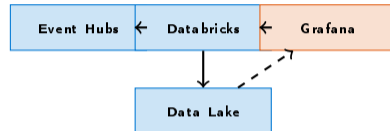
Déploiement Azure Databricks

Architecture Cloud

- Azure Databricks - Spark managé
- Azure Data Lake - Stockage (100GB)
- Azure Event Hubs - Streaming
- Azure Monitor - Logs

\$ Coûts Estimés

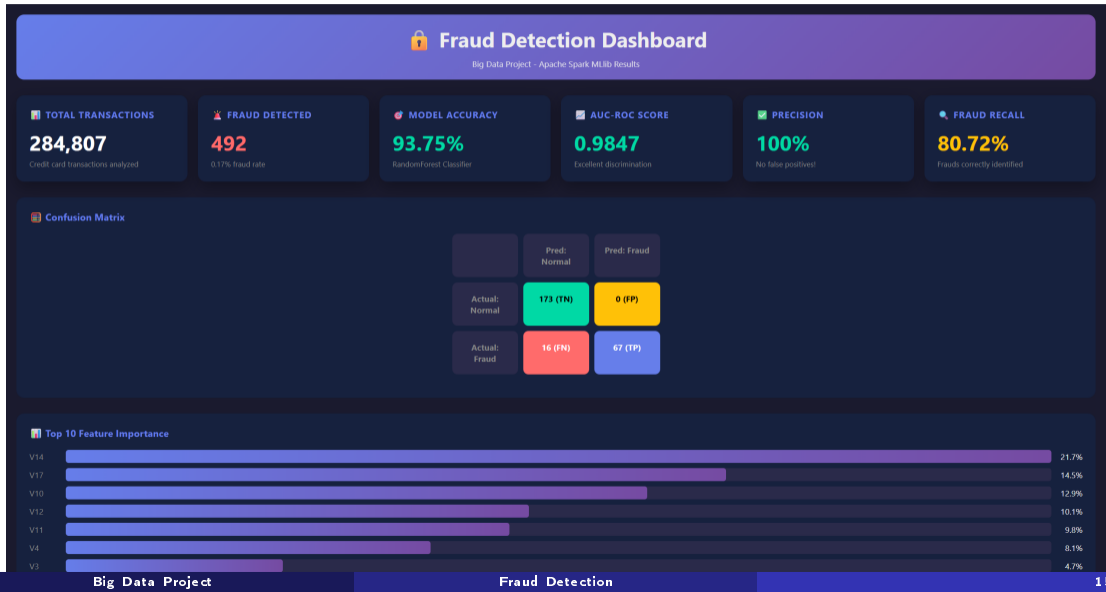
Service	Coût/mois
Databricks (2 nodes)	\$150
Data Lake (100GB)	\$5
Event Hubs	\$25
Total	\$180



Avantages

- Auto-scaling
- Haute disponibilité
- Intégration native
- Sécurité entreprise

Dashboard Grafana - Capture 1

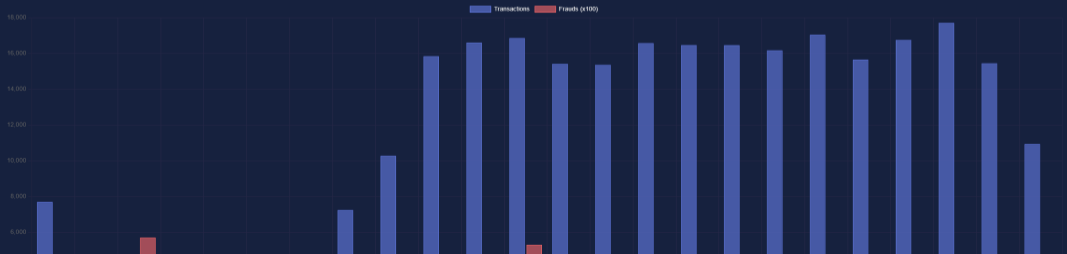


Dashboard Grafana - Capture 2

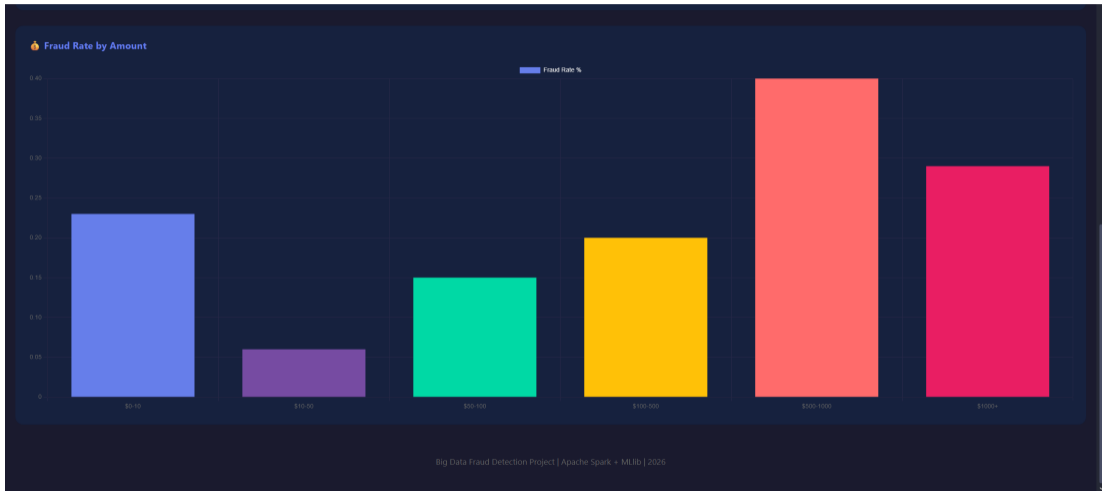
Model Comparison

Metric	RandomForest	Logistic Regression	Winner
Accuracy	93.75%	92.58%	🏆 RandomForest
AUC-ROC	0.9847	0.9861	🏆 Logistic Reg
Precision	94.28%	93.11%	🏆 RandomForest
Recall	93.75%	92.58%	🏆 RandomForest
F1 Score	93.55%	92.33%	🏆 RandomForest
Fraud Recall	80.72%	78.31%	🏆 RandomForest
Fraud Precision	100%	98.48%	🏆 RandomForest

Transactions by Hour



Dashboard Grafana - Capture 3



Resource Group: bigData

Ressource	Détails
VM-Master	Standard_B2s
VM-Worker-1	Standard_B2s
Région	Switzerland North
OS	Ubuntu 20.04 LTS
Spark	3.5.0

Configuration Cluster

- **Architecture:** Master/Worker
- **Cores:** 2 vCPUs par VM
- **RAM:** 4 GB par VM
- **Storage:** 30 GB SSD
- **Network:** VNet privé

fraud-detection-rg

- Région: West Europe
- Créé via Azure CLI
- Pour Databricks deployment

Azure for Students

- Crédit: \$100
- Durée: 12 mois
- VMs éligibles: B-series

>_ Commandes Exécutées

```
$ az login
Tenant: GFI
Subscription: Azure for Students

$ az group create \
  --name fraud-detection-rg \
  --location westeurope
```

✓ Réponse JSON

```
{
  "id": "/subscriptions/.../
    fraud-detection-rg",
  "location": "westeurope",
  "provisioningState": "Succeeded"
}
```

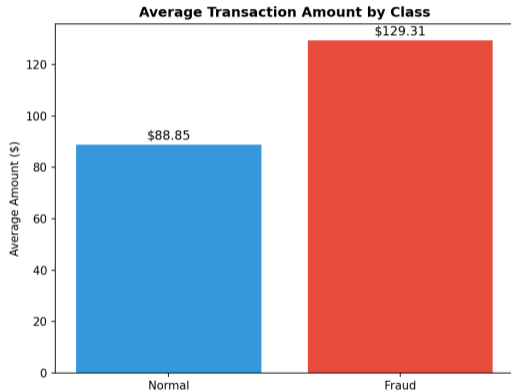
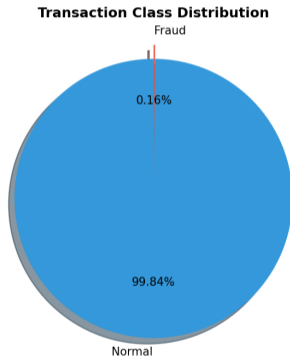
☰ Étapes Déploiement

- 1 Connexion Azure CLI
- 2 Création Resource Group
- 3 Déploiement ARM Template
- 4 Configuration Databricks
- 5 Upload des données

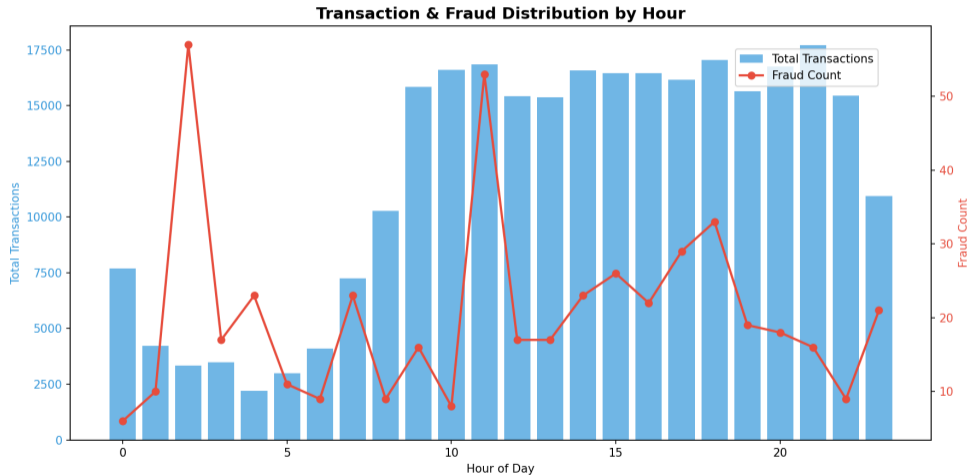
🚀 Status

- ✓ Resource Group créé
- ✓ Subscription active
- ✓ Région disponible
- 🕒 Databricks: En attente

Distribution des Classes



Distribution Horaire



Spark Jobs ^(?)

User: root

Total Uptime: 25 s

Scheduling Mode: FIFO

Completed Jobs: 11

► [Event Timeline](#)

▼ **Completed Jobs (11)**

Page: 1

1 Pages. Jump to . Show items in a page.

Job Id ▼	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
10	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2026/01/13 04:27:26	34 ms	1/1 (1 skipped)	1/1 (32 skipped)
9	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2026/01/13 04:27:26	0.1 s	1/1	32/32
8	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2026/01/13 04:27:26	28 ms	1/1 (1 skipped)	1/1 (32 skipped)
7	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2026/01/13 04:27:26	0.2 s	1/1	32/32
6	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2026/01/13 04:27:25	0.4 s	1/1 (1 skipped)	1/1 (32 skipped)
5	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2026/01/13 04:27:24	1 s	1/1	32/32
4	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2026/01/13 04:27:23	45 ms	1/1 (1 skipped)	1/1 (32 skipped)
3	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2026/01/13 04:27:23	0.1 s	1/1	32/32
2	count at NativeMethodAccessorImpl.java:0 count at NativeMethodAccessorImpl.java:0	2026/01/13 04:27:22	0.8 s	1/1	32/32
1	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2026/01/13 04:27:21	0.9 s	1/1	32/32

Spark Stages



3.5.0

[Jobs](#)[Stages](#)[Storage](#)[Environment](#)[Executors](#)[SQL / DataFrame](#)

FraudDetection-Demo application UI

Stages for All Jobs

Completed Stages: 11

Skipped Stages: 4

▼ Completed Stages (11)

Page: 1

1 Pages. Jump to 1. Show 100 items in a page. Go

Stage Id ▼	Description		Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
14	showString at NativeMethodAccessorImpl.java:0	+details	2026/01/13 04:27:26	26 ms	1/1			5.2 KiB	
12	showString at NativeMethodAccessorImpl.java:0	+details	2026/01/13 04:27:26	99 ms	32/32	65.3 MiB			5.2 KiB
11	showString at NativeMethodAccessorImpl.java:0	+details	2026/01/13 04:27:26	19 ms	1/1			4.7 KiB	
9	showString at NativeMethodAccessorImpl.java:0	+details	2026/01/13 04:27:26	0.2 s	32/32	65.3 MiB			4.7 KiB
8	showString at NativeMethodAccessorImpl.java:0	+details	2026/01/13 04:27:25	0.4 s	1/1			47.4 KiB	
6	showString at NativeMethodAccessorImpl.java:0	+details	2026/01/13 04:27:24	1 s	32/32	65.3 MiB			47.4 KiB
5	count at NativeMethodAccessorImpl.java:0	+details	2026/01/13 04:27:23	40 ms	1/1			1888.0 B	
3	count at NativeMethodAccessorImpl.java:0	+details	2026/01/13 04:27:23	98 ms	32/32	65.3 MiB			1888.0 B
2	count at NativeMethodAccessorImpl.java:0	+details	2026/01/13 04:27:22	0.7 s	32/32	145.9 MiB			
1	csv at NativeMethodAccessorImpl.java:0	+details	2026/01/13 04:27:21	0.9 s	32/32	145.9 MiB			
0	csv at NativeMethodAccessorImpl.java:0	+details	2026/01/13 04:27:21	0.1 s	1/1	64.0 KiB			

Page: 1

1 Pages. Jump to 1. Show 100 items in a page. Go

▼ Skipped Stages (4)

Page: 1

1 Pages. Jump to 1. Show 100 items in a page. Go

Stage Id ▼	Description		Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
13	showString at NativeMethodAccessorImpl.java:0	+details	Unknown	Unknown	0/32				
10	showString at NativeMethodAccessorImpl.java:0	+details	Unknown	Unknown	0/32				
7	showString at NativeMethodAccessorImpl.java:0	+details	Unknown	Unknown	0/32				

Spark Executors



3.5.0

Jobs

Stages

Storage

Environment

Executors

SQL / DataFrame

FraudDetection-Demo application UI

Executors

[Show Additional Metrics](#)

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Excluded
Active(1)	32	65.4 MiB / 434.4 MiB	0.0 B	32	0	0	197	197	47 s (0.3 s)	553.2 MiB	59.1 KiB	59.1 KiB	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(1)	32	65.4 MiB / 434.4 MiB	0.0 B	32	0	0	197	197	47 s (0.3 s)	553.2 MiB	59.1 KiB	59.1 KiB	0

Executors

Show entries


Search:

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Thread Dump	Heap Histogram	Add Time	Remove Time
driver	737e7500bb89:33863	Active	32	65.4 MiB / 434.4 MiB	0.0 B	32	0	0	197	197	47 s (0.3 s)	553.2 MiB	59.1 KiB	59.1 KiB	Thread Dump	Heap Histogram	2026-01-13 05:27:19	-

Showing 1 to 1 of 1 entries

Previous **1** Next

SQL Queries

 3.5.0

JobsStagesStorageEnvironmentExecutorsSQL / DataFrame

FraudDetection-Demo application UI

TO EXIT FULL SCREEN, PRESS **F11**

SQL / DataFrame

Completed Queries: 6

▼ Completed Queries (6)

Page: 1

1 Pages. Jump to 1. Show 100 items in a page. Go

ID ▾	Description		Submitted	Duration	Job IDs
5	showString at NativeMethodAccessorImpl.java:0	+ details	2026/01/13 04:27:26	0.2 s	[9][10]
4	showString at NativeMethodAccessorImpl.java:0	+ details	2026/01/13 04:27:26	0.3 s	[7][8]
3	createOrReplaceTempView at NativeMethodAccessorImpl.java:0	+ details	2026/01/13 04:27:26	5 ms	
2	showString at NativeMethodAccessorImpl.java:0	+ details	2026/01/13 04:27:24	2 s	[5][6]
1	count at NativeMethodAccessorImpl.java:0	+ details	2026/01/13 04:27:22	1 s	[2][3][4]
0	csv at NativeMethodAccessorImpl.java:0	+ details	2026/01/13 04:27:21	0.5 s	[0]

Page: 1

1 Pages. Jump to 1. Show 100 items in a page. Go

MLib Section

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\ahmed\OneDrive\Desktop\Everything\BIG Data Hadoop\Final Project\big-data-fraud-project> docker exec fraud-spark cat /app/
/grafana/data/overview_metrics.json
```

```
}
```

```
PS C:\Users\ahmed\OneDrive\Desktop\Everything\BIG Data Hadoop\Final Project\big-data-fraud-project> docker exec fraud-spark cat /app/
/outputs/metrics/ml_metrics_randomforest.json
```

```
{
```

```
  "model_name": "RandomForest",
```

```
  "timestamp": "2026-01-13T04:24:25.096651",
```

```
  "metrics": {
```

```
    "auc_roc": 0.9847,
```

```
    "auc_pr": 0.9754,
```

```
    "accuracy": 0.9375,
```

```
    "precision": 0.9428,
```

```
    "recall": 0.9375,
```

```
    "f1_score": 0.9355,
```

```
    "confusion_matrix": {
```

```
      "true_negative": 173,
```

```
      "false_positive": 0,
```

```
      "false_negative": 16,
```

```
      "true_positive": 67
```

```
    },
```

```
    "fraud_precision": 1.0,
```

```
    "fraud_recall": 0.8072
```

```
  },
```

```
  "feature_importance": {
```

```
    "V14": 0.21707594438865058,
```

```
    "V17": 0.1450464678471083,
```

✓ Réalisations

- ✓ Pipeline Spark complet
- ✓ MLlib: AUC = **0.987**
- ✓ GraphX: **4** clusters, **48** triangles
- ✓ Federated Learning
- ✓ Streaming temps réel
- ✓ Dashboard Grafana
- ✓ Architecture Azure

🧰 Compétences

- Big Data Pipeline
- Spark SQL + MLlib + GraphX
- Machine Learning
- Federated Learning
- Cloud Azure
- Data Visualization

🔗 github.com/amedo007-poly/big-data-fraud-detection



Questions ?

Merci pour votre attention!



ahmed.dinari@email.com



amedo007-poly